

# Local Blur Mapping: Exploiting High-Level Semantics by Deep Neural Networks

Kede Ma  
University of Waterloo  
k29ma@uwaterloo.ca

Huan Fu  
University of Technology Sydney  
Huan.Fu@student.uts.edu.au

Tongliang Liu  
University of Technology Sydney  
tliang.liu@gmail.com

Zhou Wang  
University of Waterloo  
Z.Wang@ece.uwaterloo.ca

Dacheng Tao  
University of Sydney  
dacheng.tao@gmail.com

## Abstract

The human visual system excels at detecting local blur of visual images, but the underlying mechanism is mysterious. Traditional views of blur such as reduction in local or global high-frequency energy and loss of local phase coherence have fundamental limitations. For example, they cannot well discriminate flat regions from blurred ones. Here we argue that high-level semantic information is critical in successfully detecting local blur. Therefore, we resort to deep neural networks that are proficient in learning high-level features and propose the first end-to-end local blur mapping algorithm based on a fully convolutional network (FCN). We empirically show that high-level features of deeper layers indeed play a more important role than low-level features of shallower layers in resolving challenging ambiguities for this task. We test the proposed method on a standard blur detection benchmark and demonstrate that it significantly advances the state-of-the-art (ODS F-score of 0.853). In addition, we explore the use of the generated blur map in three applications, including blur region segmentation, blur degree estimation, and blur magnification.

## 1. Introduction

Blur is one of the most common image degradations that arises from a number of sources, including atmospheric scatter, camera shake, defocus, and object motion. It is also manipulated by photographers to create visually pleasing effect that draws attention to humans/objects of interest. Given a natural photographic image, the goal of “local blur mapping” is to label every pixel as either blurred or not, resulting in a blur map. Local blur mapping is an important component in many image processing and com-

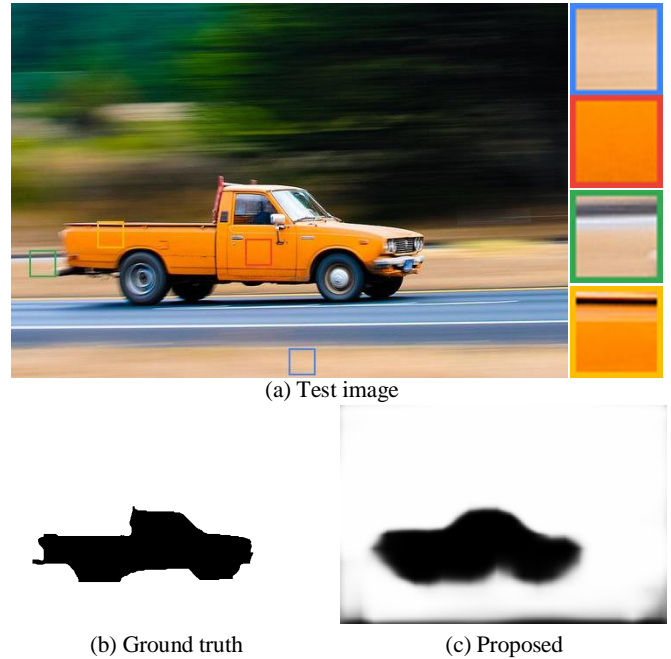
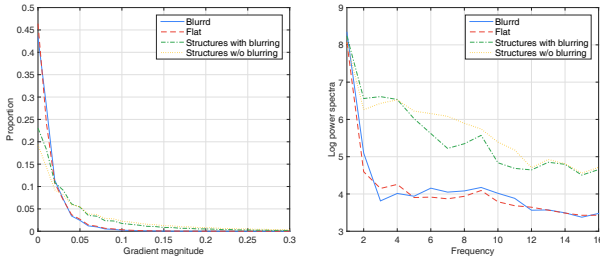


Figure 1. Challenges in local blur mapping. The pairs of (blue, red) and (green, yellow) framed patches appear to be similar in terms of local structural features and complexities, making it difficult for most local feature-based approaches to identify blur. Only semantic information can help make the distinction. The test image is from the blur detection benchmark [28].

puter vision systems. For image quality assessment, blur is an indispensable factor that contributes to perceptual image quality [33]. For example, the worst quality images that are scored by human subjects in the LIVE Challenge database [12] are mainly caused by motion and/or out-of-focus blur. For object detection, the identified blurred regions can be excluded for efficient region proposal and robust object localization [20, 26]. Other applications that



(a) Local gradient statistics (b) Local power spectral slopes

Figure 2. Traditional low-level features fail to distinguish between flat and blurry regions, and structures with and without blurring. Although the gradient distribution of blurred patches exhibits a sharp peak at zero and a less heavy tail, which is distinctive from structured patches, it cannot tell for example whether structured patches undergo blurring or not. Similar phenomena have been observed using local power spectral slopes. We extract patches from 100 images in the blur detection benchmark [28] to draw (a) and use the four patches in Fig. 1 to draw (b).

may benefit from local blur mapping include image restoration [7, 8], photo editing [2], depth recovery [23, 29], and image segmentation [10, 25].

The human visual system (HVS) is good at identifying which parts of an image appear blurred with amazing speed [35], but the underlying mechanism is not well understood. A traditional view of blur is that it reduces the energy (either globally or locally) at high frequencies. Several low-level features have been hand-crafted to exploit this observation. Among those, power spectral slopes [19, 28] and image gradient distributions [18, 19, 28] are representative. Another view is that blur results from the disruption of the local phase coherence at precisely localized features (e.g., step edges) and a coarse-to-fine phase prediction may serve as an indication of blur [34]. Nearly all previous local blur mapping operators [5, 13, 18, 19, 28] rely on the two assumptions either explicitly or implicitly, but achieve limited success. In particular, they do not resolve the ambiguity in discriminating flat and blurred regions, and they often mix up structures with and without blurring. A visual example and the associated statistical analysis are shown in Fig. 1 and Fig. 2, respectively.

In this regard, we argue that the fundamental problem in existing approaches is their ignorance to high-level semantic information in natural images, which is crucial in successfully detecting local blur. Therefore, we resort to deep convolutional neural networks (CNN) that have advanced the state-of-the-art in many high-level vision tasks such as image classification [14], object detection [26], and semantic segmentation [21]. Specifically, we develop the first fully convolutional network (FCN) [21] for end-to-end and image-to-image blur mapping [28]. By fully convolutional, we mean all the learnable filters in the network are convo-

lutional and no fully connected layers are involved. As a result, the proposed blur mapper allows input of arbitrary size, encodes spatial information thoroughly for better prediction, and maintains a relatively low computational cost. More specifically, we trim the 16-layer VGGNet [30] for image classification up to the last convolutional layer similar in [36] as our architecture. We then fine-tune the network with weights pre-trained on the semantic segmentation task [21] that contain rich high-level information about what an input image constitutes. Among the transferred hierarchical representations, we empirically show that high-level features are more important in resolving challenging ambiguities in local blur mapping, which conforms to our claim of blur perception. The proposed algorithm is tested on a standard blur detection benchmark [28] and outperforms state-of-the-art methods by a large margin.

Our contribution is three-fold. First, we provide a new perspective on blur perception, where high-level semantic information plays a critical role. Second, we show that it is possible to learn an end-to-end and image-to-image local blur mapper based on FCN [21], which addresses challenging ambiguities such as differentiating flat and blurred regions, and structures with and without blurring. Third, we explore three potential applications of the generated blur map: (1) blur region segmentation, (2) blur degree estimation, and (3) blur magnification.

## 2. Related work

The computational blur analysis is a central and long-standing problem in vision research and early works can be dated back to as early as 1960s [31]. Most researchers in this field focus on image deblurring problem that aims to restore a sharp image from a blurred one [4, 37], but blur mapping itself is little investigated. Early works on blur mapping quantify the overall blur degree of an image and cannot perform dense prediction. For example, Marziliano *et al.* analyzed the spread of the edges [22]. A similar approach was proposed in [9] by estimating the thickness of object contours. Zhang and Bergholm [38] designed a Gaussian difference signature to model the diffuseness caused by out-of-focus blur. All these methods were designed to tackle Gaussian blurred images and could not be generalized to non-Gaussian and non-uniform blur cases in the real world.

Only recently has local blur mapping become an active research topic. Rugna and Konik [6] identified blurry regions by exploiting the observation that they are more invariant to low-pass filtering. Blind deconvolution-based methods have also been investigated to segment motion-blurred [18] and defocus-blurred [16] regions. Su *et al.* [32] examined the singular value information between blurry and non-blurry regions. Chakrabarti *et al.* [5] adopted local Fourier transform to analyze directional blur. Liu *et al.* [19]

manually designed three local features represented by spectrum, gradient, and color information for blurry regions extraction and an additional local autocorrelation congruency feature for type classification. Their features have been later improved by Shi *et al.* [28] and combined with responses of learned local filters to jointly analyze blurry regions in a multi-scale fashion. All the above-mentioned methods are based on hand-crafted low-level features, which although successfully extract sharp regions, cannot tell which parts of an image are truly blurred or flat by nature.

A closely related area to blur mapping is image sharpness measurement [11, 13], which targets at extracting sharp regions from an image. The results may be combined to an overall sharpness score (global assessment) or refined to a sharpness map (local assessment). There are subtle differences between local blur mapping and sharpness assessment. For example, in sharpness assessment, flat and blurry regions can both be regarded as non-sharp, but in blur mapping, discriminating them is a must for a successful method.

### 3. Method

At a high level, we feed images of arbitrary size into an FCN and the network successively outputs a blur map of the same size, where the size mismatch is solved by in-network upsampling. Through a standard stochastic gradient descent (SGD) training procedure, our network is able to learn a complex mapping from raw image pixels to blur perception.

#### 3.1. Training and testing

**Training phase.** Given a training image set  $T = \{(\mathbf{X}_k, \mathbf{Y}_k)\}_{k=1}^K$ , where  $\mathbf{X}_k$  is the  $k$ -th raw input image and  $\mathbf{Y}_k$  is the corresponding ground truth binary blur map, our goal is to learn an FCN  $\hat{\mathbf{Y}}_k = f(\mathbf{X}_k)$  that produces a blur map with high accuracy. It is convenient to drop the subscript  $k$  without ambiguity due to the image-wise operation. We denote all layer parameters in the network as  $\mathbf{W}$ . The loss function is a sum over per-pixel losses between the prediction  $\hat{\mathbf{Y}} = \{\hat{y}_i, i = 1, \dots, |\hat{\mathbf{Y}}|\}$  and the ground truth  $\mathbf{Y} = \{y_i \in \{0, 1\}, i = 1, \dots, |\mathbf{Y}|\}$ , where  $i$  indicates the spatial coordinate. We consider the cross entropy loss

$$\mathcal{L}(\mathbf{W}) = - \sum_{i=1}^{|\mathbf{Y}|} y_i \log \Pr(\hat{y}_i = 1 | \mathbf{X}, \mathbf{W}) - (1 - y_i) \log \Pr(\hat{y}_i = 0 | \mathbf{X}, \mathbf{W}). \quad (1)$$

$\Pr(\hat{y}_i = 1 | \mathbf{X}, \mathbf{W}) = \sigma(a_i) \in [0, 1]$  is implemented by the sigmoid function  $\sigma(\cdot)$  on the  $i$ -th activation. Eq. (1) can be easily extended to account for the class imbalance situation by weighting the loss according to the proportion of positive and negative labels. Although the labels in the blur detection database [28] are mildly unbalanced (around 64% pixels are blurred), we find using the class-balanced

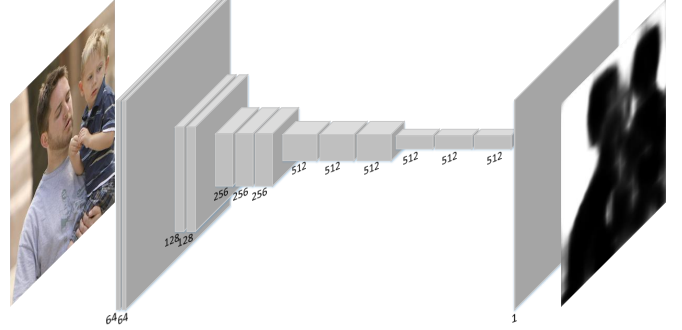


Figure 3. The trimmed VGG16 network for local blur mapping. The height and width of the cubes represent the spatial size of the filter responses depending upon the size of an input image. The depth of the cubes indicates the filter number used in each layer. Here, we omit ReLU and max pooling layers for simplicity. After the last convolutional layer (conv5\_3), our mapper performs a  $16 \times$  in-network upsampling to obtain the final blur map.

cross-entropy loss function unnecessary. Moreover, many probability distribution measures can be adopted as alternatives to the cross entropy loss, such as the fidelity loss from quantum physics [24]. We find in our experiments that the fidelity loss gives very similar performance, so we stick to the cross entropy loss throughout the paper.

**Testing phase.** After training, the layer parameters  $\hat{\mathbf{W}}$  are learned. Given a test image  $\mathbf{X}$ , we freeze the weights  $\hat{\mathbf{W}}$  and perform a standard forward pass to obtain the predicted blur map:

$$\hat{\mathbf{Y}} = f(\mathbf{X}, \hat{\mathbf{W}}). \quad (2)$$

#### 3.2. Network architecture and its alternatives

**Trimmed VGGNet.** Inspired by recent works [3, 36] that successfully fine-tune deep neural networks pre-trained on the general image classification task to the edge detection task, we adopt the 16-layer VGGNet architecture [30] but make two modifications [36]: (1) to make the network fully convolutional, we trim the network until the last convolutional layer (conv5\_3), throwing away both the 5-th pooling layer and all fully connected layers; (2) we connect an in-network upsampling layer (deconvolution) to interpolate the activations produced by the last convolutional layer so as to match the spatial size of the ground truth blur map. Fig. 3 shows the architecture. The reasons for such trimming are as follows. First, the pool5 layer produces too coarse spatial information that may pose difficulty for later interpolation. Second, instead of convolutionalizing the fully connected layers as in [21], cutting all of them significantly reduces the computational complexity with only mild loss of representation power. As a result, we speed up the computation and reduce the memory storage at both training and test stages.

We continue by discussing several more sophisticated architecture designs that better combine low-level and high-

level features and that have been successfully applied to other areas of computer vision. As will be clear in Section 4.2, incorporating low-level features through these more involved architectures often impairs performance compared with the default trimmed VGGNet.

**FCNs with skip layers.** The original FCNs make use of classification nets for dense semantic segmentation [21] by transferring fully connected layers into convolutional ones. To combat the coarse spatial information in deeper layers that limits the scale of detail in the upsampled output, Long *et al.* [21] introduced skip layers that combine the responses of the final prediction layer with those of shallower layers with finer spatial information. It is straightforward to adapt this architecture to the blur mapping task by simply replacing the loss function with the cross entropy loss. We include FCN-8s, a top-performing architecture with reasonable complexity in our experiment.

**Deeply supervised nets.** To make the learning process of hidden layers direct and transparent, Lee *et al.* [17] proposed deeply supervised nets (DSN) that add side output layers to the convolutional layers in each stage. In the case of 16-layer VGGNet adopted in edge detection [36], five side outputs are produced right after conv1\_2, conv2\_2, conv3\_3, conv4\_3, and conv5\_3 layers, respectively. All side outputs are fused to a final output, whose weights are learnable. The final output together with all side outputs contribute to the loss function, which can be minimized using a standard SGD method. We include two variants of DSN: training with weighted fusion only, and training with weighted fusion and deep supervision.

## 4. Experiments

In this section, we first provide thorough implementation details on training and testing the proposed blur mapper. We then describe the experimental protocol and compare our mapper with four state-of-the-art methods. Finally, we analyze various aspects of our algorithm with an emphasis on the role of high-level features. All models are trained and tested with Caffe [15]. The codes will be made publicly available.

### 4.1. Implementations

**Data preparation.** To the best of our knowledge, the blur detection benchmark built by Shi *et al.* [28] is the only database that is publicly available for this task. It contains 1,000 images with human labelled blur regions, among which 296 are partially motion-blurred and 704 are defocus-blurred. Since we are limited by the number of training samples available in the existing benchmark, we only divide it into training and test sets. Specifically, the training set contains images with odd indices, denoted by  $D_o$  and

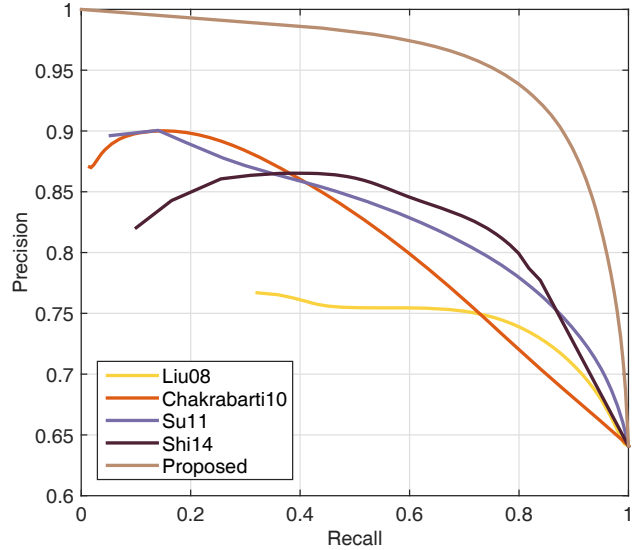


Figure 4. The precision-recall curves on  $D_e$  from the blur detection benchmark [28]. The proposed algorithm significantly boosts precisions within the entire recall range, where the improvement can be as large as 0.2.

the test set contains images with even indices, denoted by  $D_e$ . Our FCN-based mapper allows for input of arbitrary size, so we try various input sizes and find that the proposed algorithm is insensitive to input image size. We take advantage of this and resize all images to  $384 \times 384$  in order to reduce GPU memory cost and speed up training and testing. We also try to augment the training samples by randomly mirroring, rotating, and scaling the images followed by a center cropping, but this yields no noticeable improvement. Therefore, the reported results in the paper are without data augmentation.

**Optimization.** We initialize the layer parameters with weights from a full 16-layer VGGNet pre-trained on the semantic segmentation task [21] and fine-tune them by SGD with momentum. The training is regularized by weight decay (the  $l_2$  penalty multiplier set to  $5 \times 10^{-4}$ ). The learning rate is initially set to be  $2^{-10}$  and follows a polynomial decay with a power of 0.9. The learning rates for biases are doubled. The batch size is set to 3 images, and momentum to 0.9. The in-network upsampling layer is initialized and fixed to bilinear interpolation. Although those interpolation weights are learnable, the additional performance gain is marginal. The learning stops when the maximum iteration number 10,000 is reached. The final weights are used for testing.

### 4.2. Results

**Comparison with the state-of-the-art.** We compare our algorithm with four state-of-the-art methods: Liu08 [19],



Table 1. Results trained on  $D_o$ , and tested on  $D_e$  from the blur detection benchmark [28]

| Algorithm         | ODS          | OIS          | AP           |
|-------------------|--------------|--------------|--------------|
| Liu08 [19]        | 0.766        | 0.811        | 0.745        |
| Chakrabarti10 [5] | 0.758        | 0.797        | 0.757        |
| Su11 [32]         | 0.782        | 0.822        | 0.721        |
| Shi14 [28]        | 0.776        | 0.813        | 0.843        |
| Proposed          | <b>0.853</b> | <b>0.884</b> | <b>0.880</b> |

Table 2. Comparing the proposed algorithm with its various modified versions to analyze the role of high-level features

|                                | ODS          | OIS          | AP           |
|--------------------------------|--------------|--------------|--------------|
| Training from scratch          | 0.833        | 0.876        | 0.856        |
| FCN-8s                         | 0.840        | 0.874        | 0.847        |
| Fusion (w/o deep supervision)  | 0.844        | 0.877        | 0.865        |
| Fusion (with deep supervision) | <b>0.854</b> | <b>0.889</b> | 0.876        |
| Proposed                       | 0.853        | 0.884        | <b>0.880</b> |

Chakrabarti10 [5], Su11 [32], and Shi14 [28]. The quantitative performance is evaluated using the precision-recall curve.<sup>1</sup> We also summarize the performance into an overall score using three standard criteria. They are (1) optimal dataset scale (ODS) F-score obtained by finding an optimal threshold for all images in the dataset; (2) optimal image scale (OIS) F-score by averaging the best F-scores for all images; and (3) average precision (AP) by averaging over all recall levels [1].

The precision-recall curves are shown in Fig. 4. The proposed algorithm achieves the highest precisions for all the recall levels, where the improvement can be as large as 0.2. It is interesting to note that previous methods experience precision drops at low recall levels. This is no surprise because traditional methods tend to give flat regions high blur confidence and misclassify them into blurry regions even with relatively large thresholds. By contrast, our mapper automatically learns rich discriminative features, especially high-level semantics, which help accurately discriminate flat regions from blurred ones, resulting in nearly perfect precisions at low recall levels. Moreover, our method exhibits a less steep decline at the middle recall range [0.2, 0.8]. This may result from the accurate classification of structures with and without blurring. Table 1 lists the ODS, OIS, and AP results, from which we observe that our algorithm significantly advances the state-of-the-art by a large margin with an ODS F-score of 0.853.

To better investigate the effectiveness of the proposed algorithm at detecting local blur, we show some blur maps generated by the proposed algorithm and compare them with those by the most competitive methods Su11 [32] and Shi14 [28] in Fig. 5. Our algorithm is able to robustly detect

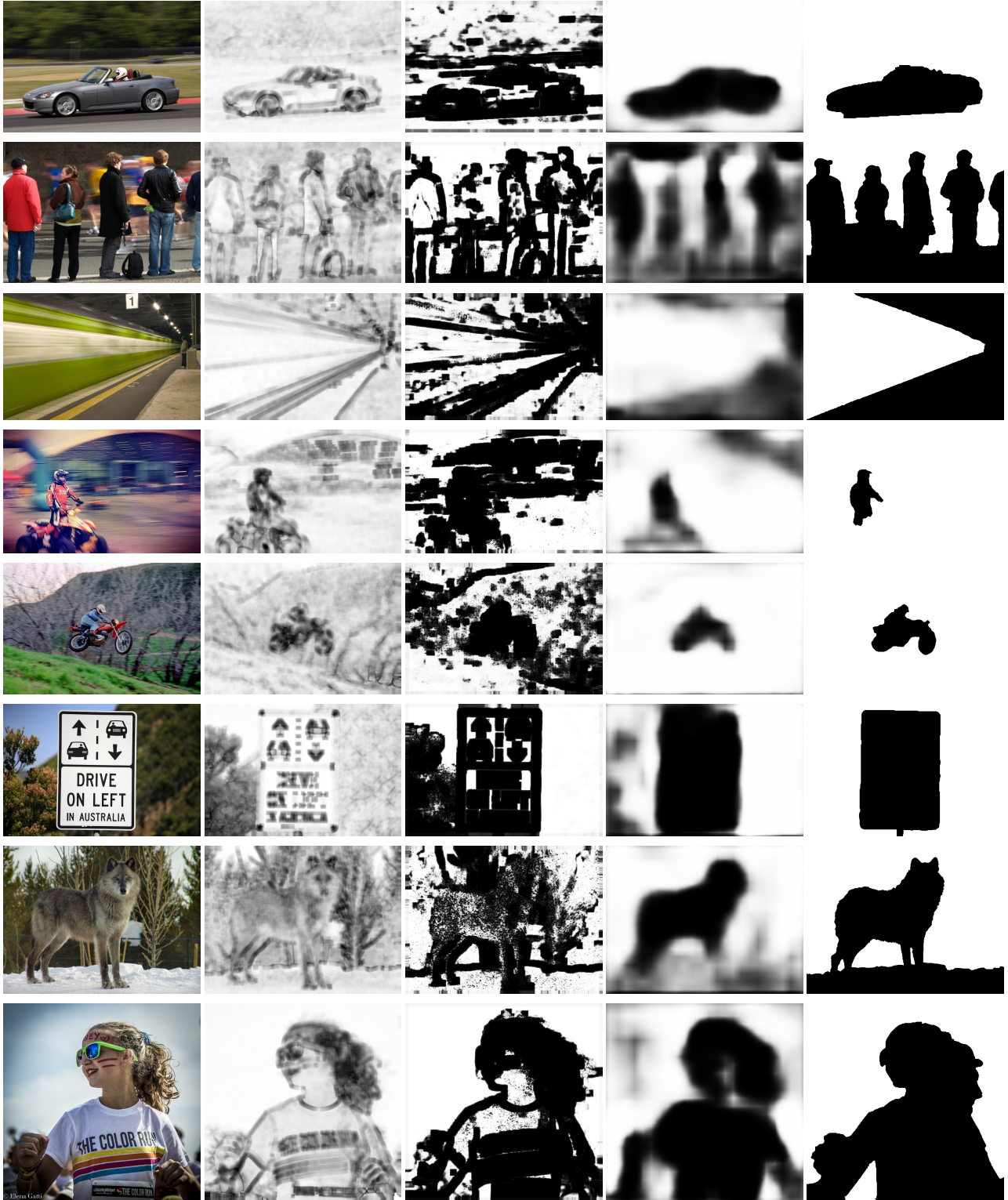
local blur from complex foreground and background. First, it handles well blur regions across different scales from the small motorcycle (in the 5-th row) to the big girl (in the 8-th row). Second, it is capable of identifying flat regions such as the car body (the first row), clothes (in the 2-nd and 8-th rows), floors (in the 3-rd row), and the road sign (in the 6-th row) as non-blurry. Third, it is barely affected by strong structures after blurring and labels those regions correctly. All of these stand in stark contrast with previous methods, which mix flat and blurry regions with high probability, and are severely biased by strong structures after blurring. Moreover, our algorithm labels images with high confidence. Nearly 40% of the pixels in the test images have predicted values either larger than 0.9 (blurry) or smaller than 0.1 (non-blurry).

**The role of high-level features.** We conduct a series of experiments to show that the learned high-level features play a crucial role in our blur mapper. We first train our FCN from scratch without using the initialization from a pre-trained net that contains rich high-level semantic information already. The results shown in the first row of Table 2 are unsatisfactory, which is expected since bad initialization can stall learning due to the instability of gradients in deep nets. By contrast, a more informative initialization with respect to the blur mapping task (in this case from semantic segmentation) is likely to guide SGD to find better local minima and results in a more meaningful blur map (Fig. 6).

We then investigate more advanced network architectures that make better use of low-level features at shallower layers, including FCN with skip layers (FCN-8s) [21], weighted fusion of side outputs [36], and weighted fusion of side outputs and deep supervision [36] (DSN). The results are shown in Table 2 and Fig. 6. We observe that although incorporating low-level features produces blur maps with somewhat finer spatial information, it voids the benefits of high-level features and results in erroneous and non-uniform blur assignment. This is expected because low-level features mainly contain edge information of an input image and do not help blur detection much. FCN-8s [21] that treats low-level and high-level features with equal importance impairs the performance most. The weighted fusion scheme without deep supervision learns to assign importance weights to the side outputs. It turns out that the side outputs generated by deeper convolutional layers are weighted heavier than those by shallower layers.<sup>2</sup> We observe slightly performance improvement over FCN-8s as expected. The weighted fusion scheme with deep supervision directly regularizes low-level features using the ground truth and delivers slightly better performance in terms of ODS and OIS than the proposed approach. In summary, the

<sup>1</sup>We draw the precision-recall curve by concatenating all test images into one vector rather than averaging the curves of all test images.

<sup>2</sup>The learned fusion weights of the five side outputs from shallow to deep layers are [0.052, 0.149, 0.160, 0.273, 0.384], respectively.



(a) Test image

(b) Su11 [32]

(c) Shi14 [28]

(d) Proposed

(e) Ground truth

Figure 5. Representative blur mapping results on the blur detection benchmark [28]. The proposed mapper shows a clear advantage in terms of accuracy over Su11 [32] and Shi14 [28], and is more consistent with the ground truths.

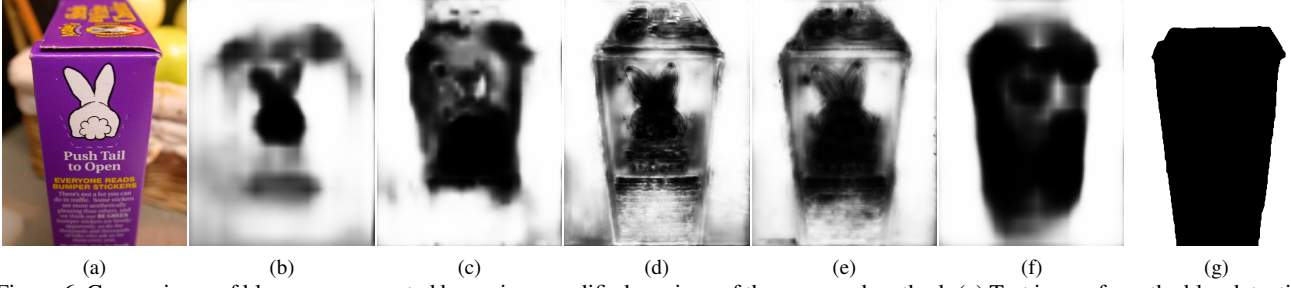


Figure 6. Comparisons of blur maps generated by various modified versions of the proposed method. (a) Test image from the blur detection benchmark [28]; (b) Training from scratch; (c) Training with FCN and skip layers [21] (FCN-8s); (d) Training with weighted fusion only [36]; (e) Training with weighted fusion and deep supervision [36] (DSN); (f) Proposed; (g) Ground truth.

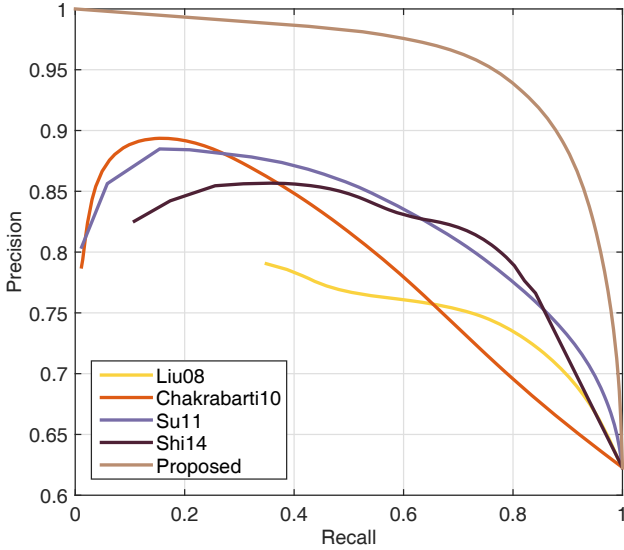


Figure 7. The precision-recall curves on  $D_o$  from the blur detection benchmark [28]. Our mapper achieves similar superior performance when using  $D_e$  for training, indicating its independence of specific training sets.

Table 3. Results trained on  $D_e$ , and tested on  $D_o$  from the blur detection benchmark [28]

| Algorithm         | ODS          | OIS          | AP           |
|-------------------|--------------|--------------|--------------|
| Liu08 [19]        | 0.753        | 0.803        | 0.749        |
| Chakrabarti10 [5] | 0.741        | 0.788        | 0.741        |
| Su11 [32]         | 0.775        | 0.814        | 0.712        |
| Shi14 [28]        | 0.765        | 0.804        | 0.831        |
| Proposed          | <b>0.852</b> | <b>0.885</b> | <b>0.876</b> |

proposed default architecture that solely interpolates from high-level feature activations achieves comparable performance with its most complicated variant DSN and even ranks the best in terms of AP. This manifests the central role of high-level features in local blur mapping.

**Independence of training sets.** We show that the perfor-



(a) Shi14 [28] (b) Proposed  
Figure 8. The blur region segmentation results.

mance of our mapper is independent of specific training sets by changing the role of the training and test sets in our setting. In other words, we train the net on  $D_e$  and test on  $D_o$  this time. We observe in the Fig. 7 and Table 3 that similar superior performance has been achieved in terms of the precision-recall curve, ODS, OIS, and AP. This verifies that our mapper does not rely on any specific training set as long as it is diverse enough to cover the natural scenes and various causes of blur.

**More training data.** Deep learning algorithms have dominated many computer vision tasks, at least in part due to the availability of large amounts of labelled data for training. However, in local blur mapping, we are limited by the number of training images available in the existing benchmark. Here we want to explore whether more training data further benefit our algorithm. To do this, we randomly sample 400 images from  $D_e$ , incorporate them into  $D_o$ , and evaluate the result on the remaining 100 images. The result averaged over 5 such trials is reported. We observe that by adding more training images, performance improves from  $ODS = 0.862$  to  $ODS = 0.869$ . This indicates that we may



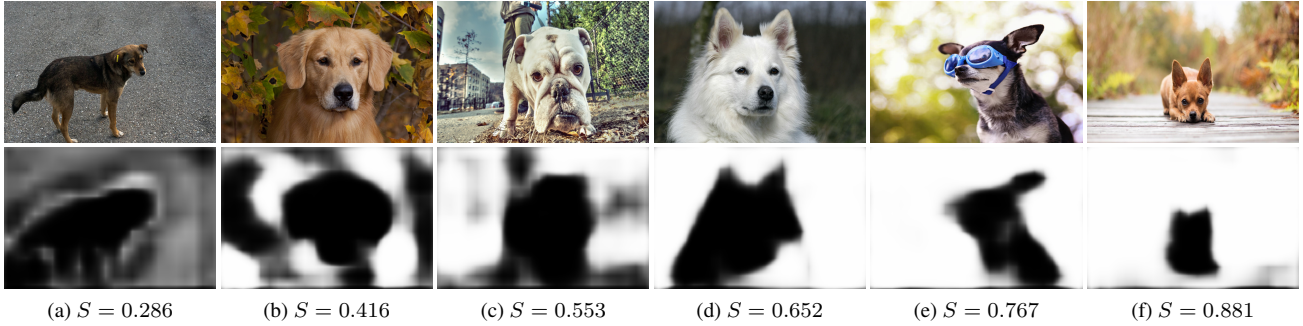


Figure 9. The overall blur degree estimation based on our blur maps. The dog images in the first row are ranked from left to right based on the estimated blur degrees as the average of the corresponding blur maps in the second row.



Figure 10. The blur magnification results.

further boost the performance and enhance the robustness of the proposed algorithm by training it with a larger dataset.

### 4.3. Applications

In this subsection, we explore three potential applications that benefit from the blur map generated by our mapper: (1) blur region segmentation, (2) blur degree estimation, and (3) blur magnification.

**Blur region segmentation.** Many interactive image segmentation tools require users to manually create a mask to roughly indicate what parts belong to foreground and background. The blur map produced by our algorithm provides a useful mask to initialize segmentation without human intervention. Here we adopt GrabCut [27] and set pixels with blur confidence  $[0, 0.1)$ ,  $[0.1, 0.5)$ ,  $[0.5, 0.9)$ , and  $[0.9, 1]$  as foreground, probable foreground, probable background, and background, respectively. We compare our results with Shi14 [28] in Fig. 8 and observe that our method does a better job in segmenting images into blur and clear regions due to more accurate blur maps. By contrast, Shi14 [28] is biased by flat regions in the foreground and structures with blurring in the background and segments images into non-connected parts.

**Blur degree estimation.** Our blur map can also serve as an estimation of the overall blur degree of an image. To do that, we design a simple blur degree measure  $S$  of an image as the average value of its corresponding blur map. Fig. 9 shows a set of dog pictures ranked from left to right with the increasing of  $S$ , from which we can see that our mapper robustly extracts blurred regions with high confidence and that the ranking results are in close agreement with human perception.

**Blur magnification.** With the extracted blurred regions, it is easy to increase defocus for blur magnification [2]. We compare the result using the blur map generated by the proposed algorithm with that of Shi14 [28] in Fig. 10. It is clear that our method is barely affected by the structures with blurring and delivers a more perceptually consistent result with smooth transitions from clear to blur regions.

## 5. Conclusion

In this paper, we shed some light on visual blur perception of natural scenes, emphasizing on the importance of high-level semantic information. We opt for CNN as a proper tool to explore high-level features, and develop the first end-to-end and image-to-image blur mapping operator based on FCN. The proposed algorithm significantly out-



performs previous methods and successfully resolves challenging ambiguities such as flat and blurred regions distinction. In the future, it remains to be seen how the low-level features and high-level semantics interplay with each other in the visual system and how they can be used to predict visual blur perception.

## Acknowledgements

The authors would like to thank Dr. Wangmeng Zuo, Dongwei Ren for deeply insightful comments, Kai Zhang and Faqiang Wang for sharing their expertise on CNN.

## References

- [1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. [5](#)
- [2] S. Bae and F. Durand. Defocus magnification. *Computer Graphics Forum*, 26(3):571–579, 2007. [2](#), [8](#)
- [3] G. Bertasius, J. Shi, and L. Torresani. DeepEdge: A multi-scale bifurcated deep network for top-down contour detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4380–4389, 2015. [3](#)
- [4] M. Cannon. Blind deconvolution of spatially invariant image blurs with phase. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(1):58–63, 1976. [2](#)
- [5] A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2512–2519, 2010. [2](#), [5](#), [7](#)
- [6] J. Da Rugna and H. Konik. Automatic blur detection for meta-data extraction in content-based retrieval context. In *SPIE Internet Imaging*, pages 285–294, 2003. [2](#)
- [7] S. Dai and Y. Wu. Removing partial blur in a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2544–2551, 2009. [2](#)
- [8] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin. Accurate blur models vs. image priors in single image super-resolution. In *IEEE International Conference on Computer Vision*, pages 2832–2839, 2013. [2](#)
- [9] J. H. Elder and S. W. Zucker. Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):699–716, 1998. [2](#)
- [10] P. Favaro and S. Soatto. A variational approach to scene reconstruction and image segmentation from motion-blur cues. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 631–637, 2004. [2](#)
- [11] R. Ferzli and L. J. Karam. A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB). *IEEE Transactions on Image Processing*, 18(4):717–728, 2009. [3](#)
- [12] D. Ghadiyaram and A. C. Bovik. Massive online crowdsourced study of subjective and objective picture quality. *IEEE Transactions on Image Processing*, 25(1):372–387, 2016. [1](#)
- [13] R. Hassen, Z. Wang, and M. M. Salama. Image sharpness assessment based on local phase coherence. *IEEE Transactions on Image Processing*, 22(7):2798–2810, 2013. [2](#), [3](#)
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. [2](#)
- [15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, pages 675–678, 2014. [4](#)
- [16] L. Kovacs and T. Sziranyi. Focus area extraction by blind deconvolution for defining regions of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1080–1085, 2007. [2](#)
- [17] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *International Conference on Artificial Intelligence and Statistics*, pages 562–570, 2015. [4](#)
- [18] A. Levin. Blind motion deblurring using image statistics. In *Advances in Neural Information Processing Systems*, pages 841–848, 2006. [2](#)
- [19] R. Liu, Z. Li, and J. Jia. Image partial blur detection and classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. [2](#), [4](#), [5](#), [7](#)
- [20] S. Liu, C. Lu, and J. Jia. Box aggregation for proposal decimation: Last mile of object detection. In *IEEE International Conference on Computer Vision*, pages 2569–2577, 2015. [1](#)
- [21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. [2](#), [3](#), [4](#), [5](#), [7](#)
- [22] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi. A no-reference perceptual blur metric. In *IEEE International Conference on Image Processing*, pages 57–60, 2002. [2](#)
- [23] G. Mather. The use of image blur as a depth cue. *Perception*, 26(9):1147–1158, 1997. [2](#)
- [24] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010. [3](#)
- [25] X. Qi, J. Shi, S. Liu, R. Liao, and J. Jia. Semantic segmentation with object clique potential. In *IEEE International Conference on Computer Vision*, pages 2587–2595, 2015. [2](#)
- [26] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. [1](#), [2](#)
- [27] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004. [8](#)
- [28] J. Shi, L. Xu, and J. Jia. Discriminative blur detection features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2972, 2014. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [29] J. Shi, L. Xu, and J. Jia. Just noticeable defocus blur detection and estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 657–665, 2015. [2](#)
- [30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representation*, 2015. [2](#), [3](#)

- [31] D. Slepian. Restoration of photographs blurred by image motion. *Bell System Technical Journal*, 46(10):2353–2362, 1967. [2](#)
- [32] B. Su, S. Lu, and C. L. Tan. Blurred image region detection and classification. In *ACM International Conference on Multimedia*, pages 1397–1400, 2011. [2](#), [5](#), [6](#), [7](#)
- [33] Z. Wang and A. C. Bovik. Modern image quality assessment. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 2(1):1–156, 2006. [1](#)
- [34] Z. Wang and E. P. Simoncelli. Local phase coherence and the perception of blur. In *Advances in Neural Information Processing Systems*, pages 1435–1442, 2004. [2](#)
- [35] M. A. Webster, M. A. Georgeson, and S. M. Webster. Neural adjustments to image blur. *Nature Neuroscience*, 5(9):839–840, 2002. [2](#)
- [36] S. Xie and Z. Tu. Holistically-nested edge detection. In *IEEE International Conference on Computer Vision*, pages 1395–1403, 2015. [2](#), [3](#), [4](#), [5](#), [7](#)
- [37] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *European Conference on Computer Vision*, pages 157–170, 2010. [2](#)
- [38] W. Zhang and F. Bergholm. Multi-scale blur estimation and edge type classification for scene analysis. *International Journal of Computer Vision*, 24(3):219–250, 1997. [2](#)